

Improving shortest paths in the Delaunay triangulation

Manuel Abellanas¹, Mercè Claverol², Gregorio Hernández¹, Ferran Hurtado³, Vera Sacristán³, Maria Saumell³, Rodrigo I. Silveira³

¹ Facultad de Informática, Universidad Politécnica de Madrid
{mabellanas,gregorio}@fi.upm.es

² Departament de Matemàtica Aplicada IV, Universitat Politècnica de Catalunya
merce@ma4.upc.edu

³ Departament de Matemàtica Aplicada II, Universitat Politècnica de Catalunya
{ferran.hurtado,vera.sacristan,maria.saumell,rodrigo.silveira}@upc.edu

Abstract. We study a problem about shortest paths in Delaunay triangulations. Given two nodes s, t in the Delaunay triangulation of a point set P , we look for a new point p that can be added, such that the shortest path from s to t in the Delaunay triangulation of $P \cup \{p\}$ improves as much as possible. We study properties of the problem and give efficient algorithms to find such a point when the graph-distance used is Euclidean and for the link-distance. Several other variations of the problem are also discussed.

1 Introduction

There are many applications involving communication networks where the underlying physical network topology is not known, too expensive to compute, or there are reasons to prefer to use a logical network instead. An example of an area where this occurs is ad-hoc networks, where nodes can communicate with each other when their distance is below some threshold. Even though the routing is done locally, to avoid broadcasting to all neighbors every time a packet needs to be sent, some logical network topology and routing algorithm must be used.

The Delaunay triangulation is often used to model the overlay topology [4, 6] due to several advantages: it provides locality, scales well, and in general avoids high-degree vertices, which can create serious bottlenecks. In addition, several widely-used *localized* routing protocols guarantee to deliver the packets when the underlying network topology is the Delaunay triangulation [3]. Furthermore, there are localized routing protocols based on the Delaunay triangulation where the total distance traveled by any packet is never more than a small constant factor times the network distance between source and destination (e.g., [3]). Since the Delaunay triangulation is known to be a spanner [5], in the case of geometric networks this guarantees that all packets travel at most a constant times the minimum travel time.

In this paper we consider the problem of improving a geometric network, with a Delaunay triangulation topology, by augmenting it with additional nodes. In particular, we aim at improving the shortest path on the Delaunay network between two given nodes s and t . Adding new nodes to a Delaunay network produces changes in the network topology that can result in equal, shorter, or longer shortest paths between s and t (an example where adding a point shortens the path is shown in Figure 1, left).

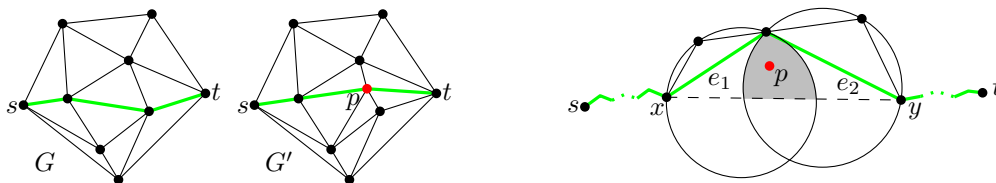


FIGURE 1. Left: shortest path between s and t before and after adding p ; the latter is shorter. Right: any point inserted in the shaded region, like p , improves $SP(s, t)$, shown in green.

We restrict ourselves to the scenario where at most one node can be added to the network, which can be placed anywhere on the plane. The goal is to find a location for the new node that improves the shortest path between s and t as much as possible. We are not aware of any previous work on this problem.

Notation. The input to the problem is a set of n points $P = \{p_1, \dots, p_n\}$, and two distinguished points $s, t \in P$. The points represent the locations of the network nodes.

We will use G to denote the Delaunay graph of P . Thus, G has the points in P as vertices, and an edge (p_i, p_j) between two vertices if and only if there is a circle through p_i, p_j that does not contain any point from P in its interior. We assume that the points in P are in general position: no three points are collinear and no four points are cocircular. Hence G represents the Delaunay triangulation of P . Moreover, we also assume that the edge $(s, t) \notin G$, otherwise the distance between s and t in G would be optimal. Note that we use G to refer to both the graph and the triangulation.

The shortest path on G between s and t will be denoted by $SP_G(s, t)$. The length of such path, defined as the sum of the Euclidean lengths of its edges, will be denoted by $|SP_G(s, t)|$, although we will omit G if possible. The straight line segment between two points x and y will be denoted by \overline{xy} , and its Euclidean length by $|\overline{xy}|$.

Finally, we will use G'_p to denote the Delaunay graph of $P \cup \{p\}$, for some $p \notin P$ (we will omit p when clear from the context).

2 Properties and observations

We begin the paper by analyzing some geometric properties of the problem.

When a new point p is inserted in P , some edges of the Delaunay triangulation might disappear and new edges, all incident to p , appear. The edges of G that are affected by the insertion of p belong to Delaunay triangles whose circumcircles contain p . In particular, all triangles in G whose circumcircle contains p get new edges in G' , connecting their vertices to p . If p is outside the convex hull of P , some additional edges might appear.

A first question that one may ask is whether it is always possible to improve a shortest path by adding one point to G . There are situations in which it is easy to obtain *some* improvement:

Lemma 2.1. *Let e_1 and e_2 be two consecutive edges of $SP_G(s, t)$. Let C_1 and C_2 be two Delaunay circles through the extremes of e_1 and e_2 , respectively. If C_1 and C_2 are not tangent and $C_1 \cap C_2$ is on the side in which the edges form the smallest angle, then the length of $SP_G(s, t)$ can always be reduced by inserting one point.*

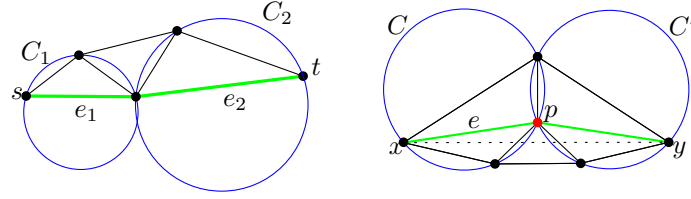


FIGURE 2. Left: example where $SP(s, t)$ cannot be improved by adding one point. Right: one of the situations described in Lemma 3.1.

Even though we omit the proof of the previous lemma, the idea can be seen in Figure 1 (right). If we insert p in the shaded region, then (x, p) and (p, y) will be Delaunay edges in G' , shortening the portion of $SP(s, t)$ between x and y . However, some shortest paths cannot be improved by adding one point, as shown in Figure 2 (left).

Lemma 2.2. *It is sometimes impossible to improve $SP_G(s, t)$ by inserting only one point.*

On the other hand, it is easy to see that two points always suffice to improve $SP(s, t)$ (details are given in [1]).

3 Finding a point that gives the maximum improvement

Next we present an algorithm that computes a point p such that $|SP_{G'_p}(s, t)|$ is minimum. The correctness of the algorithm is based on the following lemmas, proved in [1]. We use the facts that we only need to look at $O(k)$ possible candidate points, where k is the number of pairs of Delaunay circles that intersect ($k \in \Theta(n^2)$ in the worst case) and that the shortest paths from s and t need to be computed only once.

Lemma 3.1. *Let p be a point that gives the maximum improvement, and let x, y be the points in G such that $SP_{G'_p}(s, t)$ includes (x, p) and (p, y) as consecutive edges. Then p lies on the segment \overline{xy} , or on the intersection of a Delaunay circumcircle through x , and another through y .*

Lemma 3.2. *Let p be a point, and let $x \in P$ such that $(x, p) \in G'_p$. If $SP_{G'_p}(s, p)$ includes (x, p) , then $|SP_{G'_p}(s, p)| = |SP_G(s, x)| + |\overline{xp}|$. Otherwise, $|SP_{G'_p}(s, p)| \leq |SP_G(s, x)| + |\overline{xp}|$.*

Algorithm. The previous lemmas imply that in order to find an optimal point p it suffices to analyze each pair of intersecting Delaunay circles of G . We first precompute the shortest path trees from s and from t . Then we use an output-sensitive algorithm to compute all pairs of intersecting circles.

For each pair of intersecting circles (C_1, C_2) , we proceed as follows. Each circle corresponds to a Delaunay triangle from G . Let the two triangles be t_1, t_2 . For each pair of vertices $x \in t_1$ and $y \in t_2$, we first check if \overline{xy} intersects $C_1 \cap C_2$. If it does, we take p as any point on $(\overline{xy} \cap C_1 \cap C_2)$. Otherwise, we check the two points where C_1 and C_2 intersect, and use the one that gives the shortest path from x to y . See Figure 2 (right).

If the length of $SP(s, t)$ improves by using p , we update this information. In the end we output the point that gave the shortest path, if it improves over $SP_G(s, t)$, or report that no point can improve it.

The running time of the algorithm is dominated by the time needed to find all pairs of intersecting circles. Using an algorithm like Balaban's [2], we obtain an $O(n \log n + k)$ running time, with $O(n)$ space.

Theorem 3.3. *Given the Delaunay triangulation of n points, and two vertices s and t , a point whose insertion gives the maximum improvement in $SP(s, t)$ can be found in $O(n \log n + k)$ time, where k is the number of pairs of intersecting Delaunay circles.*

Related problems. We show in [1] that, based on the previous lemmas, some other related problems can also be solved efficiently. In addition, a different proximity graph can be used (e.g., Gabriel or nearest neighbor graph) by applying the general approach of partitioning the plane into regions such that inserting a point anywhere inside one region produces the same topological change to the structure.

4 Using the link-distance

An interesting variant of the problem arises when the metric used to measure distances on G is the link-distance: the length of a path is defined by its number of edges. This metric is also interesting in networking applications, since it measures the number of hops. As before, we are interested in adding one new point to G such that the link-distance between s and t is minimized as much as possible. Using a data structure based on additively-weighted Voronoi diagrams of the circle centers, this problem can be solved more efficiently than the previous one. In [1] we prove:

Theorem 4.1. *Given the Delaunay triangulation of n points, and two vertices s and t , a point whose insertion gives the maximum improvement in $SP(s, t)$, under the link-distance, can be found in $O(n \log^2 n)$ time.*

5 Discussion

We have studied the problem of adding a point to a Delaunay triangulation, such that it improves a shortest path as much as possible. In [1] we also show how to solve several other related problems. Perhaps the most intriguing question left open is whether the decision problem (Is there a point that improves $SP(s, t)$?) can be solved faster than the optimization problem.

Acknowledgments. Research initiated during the 6th Iberian Workshop on Computational Geometry, held in Aveiro. M.A. and G.H. were partially supported by project MTM2008-05043. M.C., F.H., V.S. and M.S. were partially supported by projects MTM-2009-07242 and Gen. Cat. DGR 2009SGR1040. M.A., G.H., M.C. and F.H. were also supported by project HP2008-0060. R.I.S. was supported by the Netherlands Organisation for Scientific Research (NWO).

References

- [1] M. Abellanas, M. Claverol, G. Hernández, F. Hurtado, V. Sacristán, M. Saumell, and R. I. Silveira, Improving shortest paths in the Delaunay triangulation, manuscript in preparation.
- [2] I. J. Balaban, An optimal algorithm for finding segments intersections, in *Proc. SoCG'95*, 1995, 211–219.
- [3] P. Bose and P. Morin, Online routing in triangulations, *SIAM J. Comput.*, **33** (2004), 937–951.
- [4] E. Buyukkaya and M. Abdallah, Efficient triangulation for P2P networked virtual environments, in *Proc. NetGames'08*, 2008, 34–39.
- [5] D. P. Dobkin, S. J. Friedman, and K. J. Supowit, Delaunay graphs are almost as good as complete graphs, *Discrete Comput. Geom.*, **5** (1990), 399–407.
- [6] J. Liebeherr, M. Nahas, and W. Si, Application-layer multicasting with Delaunay triangulation overlays, *IEEE J. Sel. Areas Comm.*, **20** (2002), 1472–1488.